Dependability and Architecture: An HDCP Perspective

William L Scherlis
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA
scherlis@cmu.edu

Dependability and architecture

It is generally accepted in engineering practice that dependability, like security, is best designed into a system from the outset—that certain architectural design characteristics positively correlate with overall system dependability. Dependability ("reliance that can justifiably be placed") necessarily comprises, in this respect, a broad range of attributes such as fault tolerance, robustness, code safety, safe concurrency, usability, and self-healing behavior. And, in the interests of scalability, our definition of dependability must be meaningful relative both to overall system behavior and to the behavior of internal (sub)systems with respect to clients.

For the most part, we lack a systematic scientific linking of architectural characteristics with overall dependability outcomes, and this applies to most of the attributes mentioned above. There is only preliminary literature identifying concrete hypotheses concerning "favorable" architectural characteristics. But, perhaps more frustrating, even in the presence of such hypotheses we lack the ability to measure directly the critical variables to evaluate their validity, and must instead rely on weak surrogates. In this respect even small successes in measurement can help engineers develop more prescriptive approaches to architecting and implementing high dependability systems. (The substantial premiums paid by customers, for example, for higher availability in data centers supports this claim.)

Another, perhaps greater, challenge may be described as "dependability remediation"—a rubric meant to include both the evaluation and the improvement of existing systems with respect to particular dependability attributes. How can overall dependability be evaluated, and the relative contributions be determined for various identified design decisions, and with respect to particular attributes? It is tempting to dismiss this as an almost hopelessly broad question. But some focus can be achieved by addressing specifically the development of "incremental techniques" for remediation (i.e., measurement, improvement, assurance)—in which individual actions of engineers to make local improvements yield increments of overall improvement. This incrementality is a feature of successful open source engineering practice.

The High Dependability Computing Project (HDCP) was recently initiated by NASA Ames to address some of these issues in the context of future NASA systems. The research is directed at understanding dependability issues in larger systems and developing practicable techniques for evolution and improvement. The program combines research on measurement (correlative process/product measures for various dependability attributes), assurance (analytically based dependability claims), and technological intervention (techniques for designing dependability systems, or improving the dependability of existing systems).

The HDCP is meant to be a genuinely collaborative effort—NASA systems and projects are objects of study, in order to understand the challenges of moving techniques for measurement, assurance, and improvement from laboratory into practice. A diverse portfolio of research teams and approaches in HDCP reduces the risk of engagement for NASA mission organizations engaging in testbed projects. In order to achieve the diversity of approaches and the extent of logistical support required to support the testbed projects, Carnegie Mellon is collaborating with five other universities, including University of Southern California, University of Maryland, MIT, University of Washington, and University of Wisconsin Milwaukee. In addition, the team will be augmented by researchers funded through a recently-announced solicitation from the National Science Foundation—Highly Dependable Computing and Communication Systems Research (HDCCSR)—that builds on HDCP testbed projects.

Within HDCP, for example, a number of aspects of architecture are addressed, including both architectural design and architecture implementation. Examples include self-healing architecture designs, robustness testing of internal services and the interfaces through which they are delivered, development of architectural metrics, and model-based evaluation of API compliance.

Acknowledgement. The author wishes to acknowledge support through the High Dependability Computing Program from NASA Ames cooperative agreement NCC-2-1298.